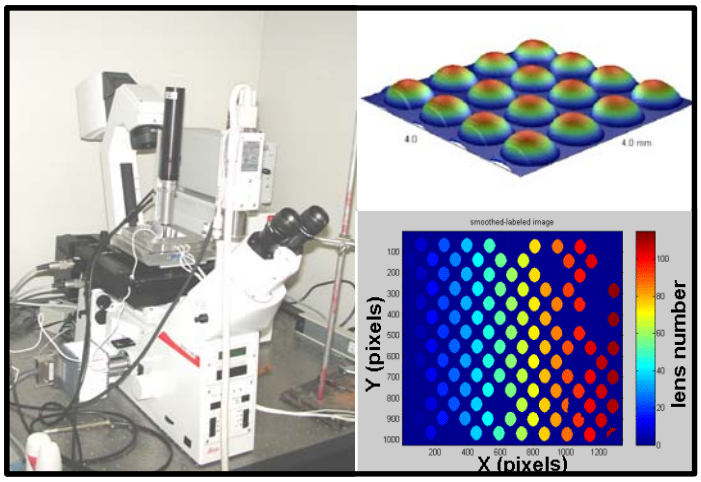


Multilens Contact Adhesion Test (MCAT)

Specifications and Operation Guidelines

For parallel testing and quantification of interfacial adhesion across combinatorial libraries.

MEASUREMENT • ANALYSIS • INFORMATICS • DESIGN • FABRICATION •

NIST Combinatorial Methods Center

Disclaimer

Certain commercial equipment, instruments, or materials are identified in this document in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

This document is intended as a basic guide for constructing an instrument but does not imply any agreement for continuing technical support. For more information, contact the NIST Combinatorial Methods Center at NCMChelp@nist.gov

This report is made available for public knowledge of NIST Combinatorial Methods Center members, but is not intended for general distribution.

Prepared by: Aaron M. Forster and Wenhua Zhang, Polymers Division, NIST.
Last Update/Edit: July 2004 (Forster)

Table of Contents

1. Purpose and structure of this document

This document is provided by the NIST Combinatorial Methods Center as a guide for design principles, construction, and operation of the MCAT instrument. First, the basic principles of this instrument will be described. Next, the components of the instrument and schemes for its construction are supplied. Here, the discussion is based upon the specific components and design of the NCMC device (see disclaimer, page 1). Next, guidelines for operating the instrument and some basic applications are outlined, including notes on LabView VIs for automation and ImagePro code for image analysis.

2. Introduction

The National Institute of Standards and Technology Combinatorial Methods Center (NCMC) was founded to provide leadership to both industry and academia involved in combinatorial and high-throughput (C&HT) methods development. The NCMC is focused on the developing C&HT methods and techniques for materials science in the polymer industry. An area of concentration for the NCMC has been the advancement of C&HT methods for adhesion measurements. To that end, we have been diligently working to develop a high-throughput approach to adhesion measurements.

This high-throughput adhesion test is based on the axisymmetric adhesion test geometry of Johnson, Kendall, and Roberts (JKR) contact adhesion test. The JKR theory describes the contact area when two elastic hemispheres are brought together under load as a function of material properties and adhesion between the materials. This test is usually conducted by compressing and decompressing two hemispheres against each other and measuring load, contact area, and displacement to provide a single adhesion measurement. Experimentally, the test geometry may also be a sphere against a flat substrate. This method may be applied in a serial manner for C&HT studies; however a serial approach results in a large amount of experimental time to completely characterize a combinatorial library. Another solution is to conduct parallel adhesion tests. In this manner, one loading and unloading cycle produces several adhesion tests and in a greater testing density than is possible with the traditional single lens techniques. This is the motivation for the development of a Multilens Combinatorial Adhesion Test (MCAT) platform. The MCAT technique, as will be shown, utilizes an array of hemispherical lenses to conduct multiple axisymmetric adhesion tests during one loading/unloading cycle. This document serves as the instrumentation document to describe the construction, software, and procedures for successfully operating the MCAT equipment.

3. Experimental Design

MCAT Equipment

The MCAT test instrument contains similar components to a non-combinatorial axisymmetric adhesion test. Figure 1 is a picture and schematic diagram of the MCAT apparatus. There are six main components: a microscope, actuator, xy positioning, manual tip/tilt stage, fiber optic displacement sensors, and software. The base platform is a sophisticated inverted microscope (Leica DMIRE) that provides both imaging and controls the xy position stage. The closed loop actuator and controller (Burleigh Instruments) provide z position control for the hemispherical lenses. The actuator is mounted to a second manual xy position stage or lens position stage and a 4 axis tip/tilt stage. The sample xy position stage (SPS) is a means to move both sample and actuator to different field of views (FOV) for the objective. The LPS allows independent movement of the actuator relative to the sample and first x-y position stage. There are two fiber optic displacement sensors employed to measure lens displacement. The instrument has the capability to mount a load cell in-line and normal to the lenses, although this feature is rarely used during multi-lens experiments. The MCAT set-up is complex, but does provide an extremely flexible platform for conducting high-throughput experiments.

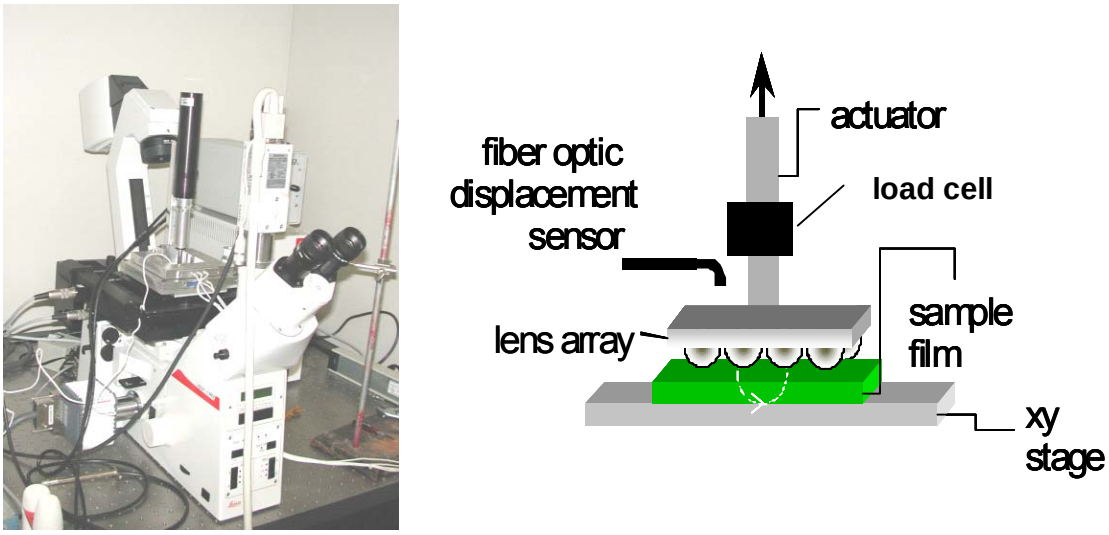


Figure 1: a) Picture of complete MCAT instrument showing the actuator and tip/tilt stage attached to the inverted microscope. b) Schematic of the MCAT axisymmetric adhesion test geometry. The lens array is attached to the nano-positioner actuator. Displacement is measured with a fiber optic displacement sensor (Philtec). The sample film is held fixed to the microscope x-y stage and the contact area is viewed through the film.

Leica Microscope

The Leica DMIRE microscope is an inverted microscope intended for use as a tool for biological studies. The scope currently has a 2.5X, 5X, 10X, and 20X objectives with fluorescence capabilities. The scope is controlled through the Image Pro software package using two integrated programs that operate within the Image Pro package. The first program is called Scope-Pro. This package controls the objectives, Hammamatsu camera, fluorescence exposure, etc. The second program is Stage-Pro and it controls the SPS. Both of these programs may be controlled with the LabView software through the use of .dll programs. Information about the function of these programs may be found in the Image Pro users manual.

Burleigh Instruments Actuator

The z movement of the hemispherical lens is performed with the Burleigh Instruments actuator. The actuator works using a series of piezoelectric clamps that can move the actuator shaft with sub-micron accuracy. LabView software interfaces with the 3200 controller to deliver motion commands to the actuator.

*LPS and tip/tilt contro*l

The lens xy position control is placed on top of the inverted microscope and the actuator is mounted atop the LPS. As indicated in the name, the LPS provides manual xy positioning of the actuator *independent* of the sample. This is allows the MCAT user to easily move to different areas of a combinatorial library for testing. Often multilens arrays are not as planar as required to contact an appreciable number of lenses within a short displacement. We have incorporated a three axis tip/tilt stage. The pitch and yaw of the actuator is controlled via three micrometers located at the corners of the tip/tilt stage as shown in Figure 2. There is a locking screw next to the third micrometer that may be used to lock the stage in place after the lenses are brought planar with the test substrate.

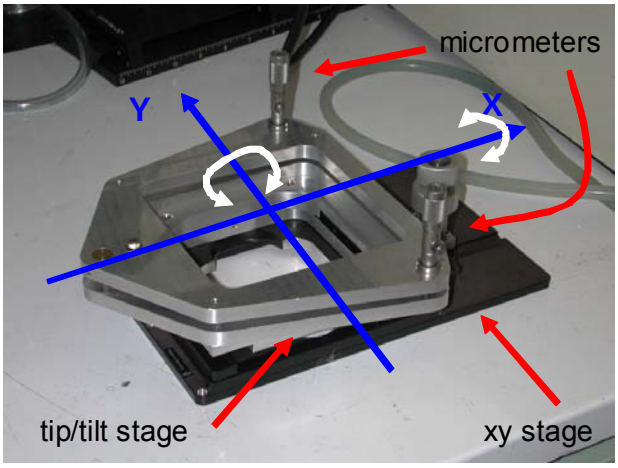
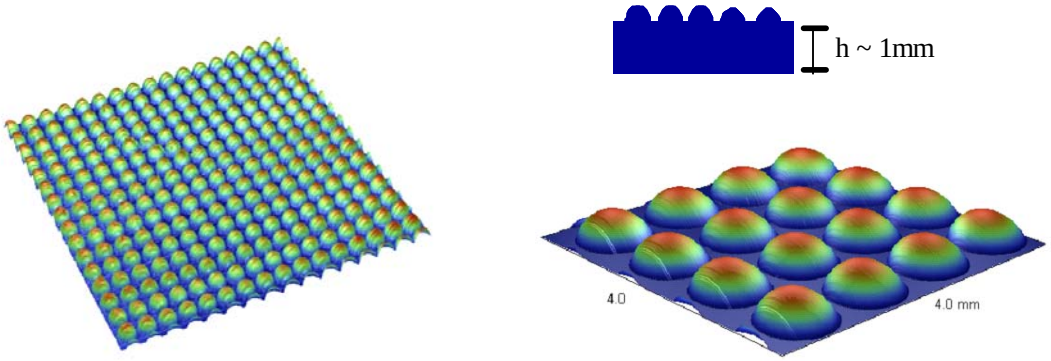


Figure 2: Tip/tilt stage for controlling the pitch and yaw of the Burleigh actuator.

Lens arrays

There are several lens arrays available to the MCAT user. Figure 3 contains an image of two of the three different arrays. At this point, we will describe the different arrays from smallest to largest. The smallest lens array available is a quartz master available for purchase from (MEMS Optical, Inc). This 40 x 40 array (1600 lenses) is approximately 1 cm^2 and each lens within the array is 300 μm in diameter, 22 μm in height and has a periodicity of 310 μm. The radius of curvature is 400 μm. These lenses provide the highest testing density currently available for MCAT experiments. Unfortunately, their elliptic profile and small height makes them slightly unsuitable for practical use as lens arrays. The second array is a larger array of 324 lenses with a footprint of 4 cm^2 and each lens within the array is 900 μm in diameter, 350 μm tall, and a periodicity of 700 mm. The radius of curvature of each lens is 400 μm. The third lens array is the most flexible in terms of investigations of combinatorial libraries. The array may be composed of up to 16 individual lenses. These lenses have a diameter of 3 mm and a radius of curvature of 1.5 mm. The advantage of these lenses is their adaptability. Arrays may be formed by gluing each lens to a glass slide in any configuration appropriate for combinatorial studies. Caution must be taken when gluing the lenses to the glass slide. The experimenter must insure that the glue will not alter the surface chemistry of the glass and that it will not contribute mechanically to the adhesion test. Also, it is much easier to modify these lenses separately and then attach them to a glass slide for a more intricate combinatorial investigation of combinatorial adhesion.



(a) **(b)**

Figure 3: Profilometer images of the multi lens arrays. a) image of a portion of the smaller lens array containing 1600 lenses over 1cm^2. b) image of a portion of the larger lens array containing 324 lenses over 3.25 cm^2. Both images are 4 mm x 4 mm slices of their respective lens arrays. The inset drawing shows the base PDMS film that supports the lenses.

Array Replication and Modification

The MCAT technique is advantageous for high throughput quantification of adhesive properties as a function of interfacial and bulk properties. In this equipment manual, we briefly discuss some techniques for creating lens arrays from masters and for modifying the surface properties of lens arrays. The simplest method to alter lens mechanical properties is by using silicone polymer precursors to create negative masters from the original lens arrays mentioned above. The elastomeric negatives may be carefully removed from the positive lens array upon curing. Figure 4 displays the steps taken to prepare a negative mold.

The silicone of choice due to ease of use and ability to release from the positive is Dow Sylgard 184. Sylgard 184 is purchased as a kit of A (polymer precursor) and B (catalyst). The silicone is thermally cured to form poly(dimethyl siloxane) (PDMS) [reference cure protocol]. Its mechanical properties are easily tuned by altering the ratio of prepolymer to catalyst. Table 1 gives the approximate modulus of PDMS created from different curing ratios. This table should serve as a guideline, as actual modulus values will depend on individual samples.

Table 1: Approximate modulus values for PDMS cured at 70 °C for 1 hour.

Mixing Ratio (mass:mass)	Modulus (MPa)
10:1	1.50
20:1	0.50
25:1	0.25

After removing the negative, a similar fabrication method may be employed to create a positive array. Materials often used to create new arrays are PDMS, Norland optically cured epoxy, and thermally cured epoxies. After curing of each positive, it recommended that the lenses are quickly washed in solvent and dried (nitrogen and vacuum). In cases where the negative and positive molds stick to each other during curing, a thin layer of Teflon AF may be spin coated to the negative mold for release. It is imperative that the molds (negative and positive) are washed to remove residual Teflon AF before further use.

Figure 4-*need to include some pictures of the arrays being made…both positive and negative.*

4. Experimental Protocol

Controls

Experimental controls: the LabView program

The LabView interface is the main program that collects images and measurement data, moves the actuator, and exports data to files. The software interface is straightforward. Figure 5 is a screen shot of the computer interface. All of the letters given below refer to letters in this figure. The conversion factors for the load and displacement sensors are shown as (a). The maximum displacement (b) and final displacement (c) are input values. The output directory for images and data is given in (d). Images are acquired through the Image Pro software via .dll instructions in Labview. The pictures are saved in sequence in the *d:\adhesion\capture* directory. The pictures are saved in a numbered sequence with a zero added after the sequence number. After running the program, the files listed in the d:\adhesion\capture directory are sequentially listed in the following manner: 00test.tif, 10test.tif, 20test.tif, 30test.tif, etc. Windows prefers to sort files by their name, therefore, the user will need to resort the directory by date in order to view the sequential progression of the files. The program runs in two modes: continuous or step. The modes are controlled by the toggle switch (e) on the panel.

In continuous mode, the actuator moves at a continuous velocity until reaching the maximum displacement. After reaching the maximum displacement, the actuator reverses direction until the final displacement. The interval between pictures may be controlled by the interval time (f). The camera acquires pictures as fast as possible when the interval time is set to zero.

In step mode, the actuator steps incrementally until the maximum displacement is reached and then the actuator reverses to the final position. After each movement the microscope objective is rastered to each of the four positions given in (g). After all four pictures are collected; the actuator is moved to the next position[†]. The directory to save images is provided using the blank line in (h), generally it is till d:\adhesion\capture. There are two wait times that are important to conducting a step test. The *picture wait time* (i) is the time (milliseconds) the computer will wait before taking a picture after the stage has reached the specified position. The *move wait time* (milliseconds) (j) is the time the computer will wait before moving the stage after the picture has been collected. The researcher should use caution when selecting wait times. If the wait times are too short, than the contact area may not be in equilibrium during image capture or the stage may not have reached the specified position before the image is collected.

[†] The position of each image is obtained by using the Image Pro stage pro interface. Manually move the stage to position and record the coordinates into the program.

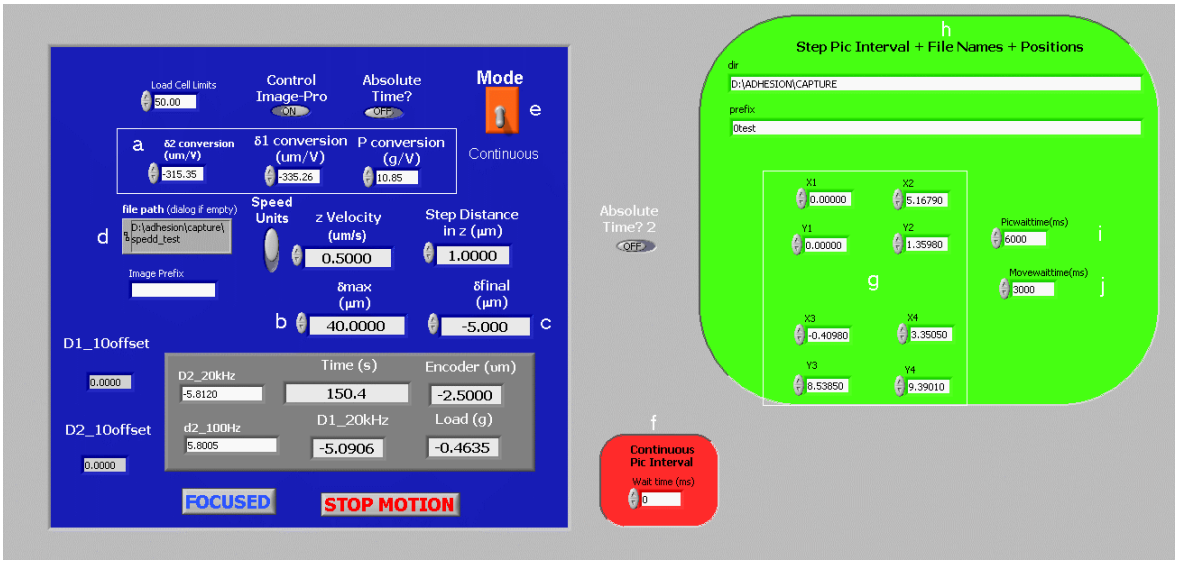


Figure 5: LabView interface. This interface is used to control the adhesion test from start to finish and collect data throughout the test.

Conducting an adhesion test: step-by-step

The following outline is provided as a guide to conducting a multilens adhesion test.

Pretest configuration

1. The user should decide on the appropriate lens array to use for investigating the substrate and prepare it accordingly.
2. The reader should refer to Figure 4 for the following substeps.
    a. Turn the power onto the Lieca DMIRE, SPS controller, Hamamatsu camera, and Burleigh controller.
    b. Open the Labview program (program name) and the Image Pro Hamamatsu icon on the desktop.
    c. After the programs load, go to Acquire tab and open the Stage Pro program.
        i. The program will prompt for the method to calibrate the stage movement. Please select *maximum travel*. WARNING: THE STAGE WILL MOVE TO ITS LIMITS. THE RESEARCHER MUST INSURE THE OBJECTIVE OR PERIPHEAL EQUIPMENT WILL NOT IMPEDE STAGE MOVEMENT.
    d. Attach the lens array to the lens holder, see Figure 4a.
    e. Place the actuator and leveling assembly onto the SPS, see Figure 4b. The stage should remain in place due to its weight.
    f. Before starting an experiment the lenses must be brought parallel to the SPS. Align the lens array with respect to the SPS. The

method of leveling the lens array is as follows: Focus onto the lens array using the microscope objective. Select four widely spaced reference points as shown in Figure 4c. The array is parallel to the SPS when all four points remain relatively in focus. The degree of error is $\pm$ 20 μm of tilt in any direction. The lens array may be rotated around the y-axis using the both micrometers in-sync and the screw. The lens array may be rotated around the x-axis using the opposing motion of the micrometers.

   g. Remove the actuator from the SPS. Place the sample onto the SPS. The sample is secured using the four screws and aluminum bars as shown in Figure 4d. Replace the actuator on top of the SPS.

3. Attach the displacement sensors to the actuator. The initial voltage should be between 2 V and 3 V.
4. Bring the lens array within a few microns of the substrate surface before starting an experiment. Close the Image Pro preview window to increase computer speed.

Test Configuration

Continuous Velocity Test

1. Enter the values for the instrument conversion factors, z-velocity, step distance, δmax, δfinal, picture interval, and file path.
2. Select the toggle switch on the control panel to the 'CONTINUOUS' position.
3. Select Image Pro control to 'ON' and Absolute time to 'OFF'.
4. Start the Labview .vi program and click the 'FOCUS' button. A pop-up window will ask you to confirm the test parameters are correct. If the parameters are correct, than push the 'OK' button.
5. The program will start running. There are several indicators that should me monitored to determine whether the equipment is behaving properly. They are as follows:
   a. D1_20k and D2_20k displacement sensors read a similar value.
   b. The computer does not slow considerably during operation.
   c. The D1_20k, D2_20k displacement sensors and the encoder reading are not off by more than 15 microns.
6. All image files and the data file are available in the specified directory (*d:\adhesion\capture*)when the test is finished.

Incremental Step Test.

1. Enter the values for the instrument conversion factors, z-velocity, step distance, δmax, δfinal, and file path.
2. Select the toggle switch on the control panel to the 'STEP' position.
3. Select Image Pro control to 'ON' and Absolute time to 'OFF'.

4. Next, select the four positions you would like to image during the step test.
    a. Open the Image Pro preview window.
    b. Use the joystick to move the SPS stage to the four positions that will be imaged during the test.
    c. The Stage Pro subprogram will provide the x and y coordinates of each position.
    d. Enter the positions in the green box on the Labview .vi interface.
    e. Enter the directory to write image files during the test.
    f. Set the picture and move wait time.
5. Start the Labview .vi program and click the 'FOCUS' button. A pop-up window will ask you to confirm the test parameters are correct. If the parameters are correct, than push the 'OK' button.
6. The program will start running. There are several indicators that should me monitored to determine whether the equipment is behaving properly. They are as follows:
    a. D1_20k and D2_20k displacement sensors read a similar value.
    b. The computer does not slow considerably during operation.
    c. The D1_20k, D2_20k displacement sensors and the encoder reading are not off by more than 15 microns.
    d. The displacement sensors to fluctuate wildly during the experiment.
7. All image files and the data file are available in the specified directory (*d:\adhesion\capture*) when the test is finished.

Image Analysis

After running the adhesion test, the images must be analyzed to determine the contact area between each lens and the substrate. This may be done in one of two ways. Images of a single contact area are analyzed using either a Matlab or Image Pro program. Images of multiple contact areas must be analyzed using the Matlab program. This section is designed to give an overview of the image analysis program and a methodology for running the programs. The actual code for the programs is provided in Appendix A. Useful hints for optimizing the code toward individual adhesion tests is provided in the comment lines.

Image Analysis Overview

A background image of the lenses out of contact is used to determine the position of each lens within the array. The background image is blurred using a Gaussian filter. Light is not uniformly reflected back into the microscope objective through the lens arrays and blurring the image smoothes non-uniformities in illumination. The blurred image is subtracted from the background image to remove dust asperities and produce an image of uniform background intensity. Next, the subtracted image is thresholded to create a new image with pixel intensities that are equal to zero above the threshold value and one below it. This leaves a binary image where the outer edge of each lens is visible as a ring with a diameter the size of the diameter of each lens. The area within each

edge is filled to produce an image that has solid circles of intensity one for each lens and a background of intensity zero outside each circle. Each circle is assigned a number and the centroid coordinates determined for use in the evaluation of experimental images.

For experimental images, the unprocessed background image (lenses out of contact) is subtracted from each image to remove the base of the lenses that does not contact the substrate. A threshold is applied to the subtracted image to create a binary image with pixel intensities that are equal to zero above the threshold value and one below it. This binary image is further dilated and eroded to highlight the contact areas, if they exist, at each lens position. Each contact area in this binary image is assigned a number and the centroid, contact area, eccentricity are calculated. Finally, the centroid positions determined from the image of the lenses out of contact is used to match the centroid positions of the experimental contact areas to track the contact behavior of each lens within the array. This method of analysis allows one to measure contact radii within ± 7.00 μm at 2.5X and ± 4.30 μm at 5X, which is comparable to manual measurements of the same contact radii that are ± 9.79 μm and ± 4.59 μm for 2.5X and 5X, respectively.

The automated image analysis lends itself to additional analysis tools such as "contact maps", described further. After each experimental image is analyzed and the relevant experimental data is recorded, the intensity of the pixels within each contact area is increased by an arbitrary intensity value of 1. As the software cycles through the subsequent images, the intensity of the pixels associated with a given lens contact area increases. A temporal contact map is created for each lens in this manner. These maps are helpful because they visually represent which lenses have been in contact with the substrate the longest during the experiment. Figure 3(b) is a contact map at different portions of the loading/unloading cycle for a lens array captured at 2.5X. The transition in color from light blue to dark red is indicative of longer contact times between the lens and substrate. Lenses that experience similar contact deformations against the substrate can be clearly illuminated utilizing these maps. This is important for determining whether lens deformation or contact time is influential in determining the work of adhesion. While contact maps provide useful information about the experimental test, more quantitative analysis is required to extract the work of adhesion.

Single contact area image analysis

Image Pro

Several macros have been written in Image Pro to measure the contact areas from each experiment. These macros will only support imaging single contact areas at a time. The procedure is as follows: build a sequence of experimental

images, subtract the background, determine the area of interest, threshold images, and run the macro to measure the contact areas.

1. After running the adhesion test, go to ACQUIRE→SEQUENCE TOOLS→MERGE FILES. A file selection window will open up.
2. Go to the image directory and sort the images by *date*. First select the last image file in the directory, hold down the shift key, and then select the first image file in the directory. The reversed selection of images forces the computer to build the sequence in the correct order. Click the 'OPEN' button after all of the image files are highlighted.
3. A sequence (movie) of all of the experimental images should now be visible within the workspace of Image Pro. A sequence control bar will also open below the original sequence. This control bar allows the user to move forward or reverse through images and select a specific sequence of images.
4. First, use the arrows located on the edge of the sequence control bar to select the first image (Figure 5).
5. Go to ACQUIRE→SEQUENCE TOOLS→EXTRACT FRAMES to extract the first image from the sequence. Save this image under the name background.tif.
6. Go to ADVANCED→BACKGROUND CORRECTION. A second window will open up. This window will specify the sequence currently open on the desktop and the file to use as a background (see step 5). Use the current image for the subtraction output. Select the ACQUIRE button and the background will be subtracted. After the background is subtracted, select the DONE button.
7. Use the slider on the sequence control bar to find the image with the largest contact area. Create a circular area of interest (AOI) around the maximum contact area.
8. Confirm the proper magnification calibration has been loaded into the Image Pro program. Go to PROCESSES→SPATIAL→CALIBRATION and select the 2.5X, 5X, or 10X lens that was used during the experiments.
9. It is now time to run the macro that will image the contact areas. The macros are located in a library directory. The directory is accessed through MACROS→LOAD. A open file directory will open. The macros are named JKR_2.5X, JKR_5X, and JKR_10X. Select the appropriate directory for the experimental objective and select open.
10. Select OK on the open window.
11. The areas in μm^2 measured with the macro are output to an EXCEL worksheet. A blank excel worksheet **MUST** be open before running the macro.
12. When you are ready to analyze the images go to MACROS→LOAD and select the macro previously loaded in step 9.
13. The macro will step through the sequence of images and measure the area of objects above the set threshold level (red outlines). The largest

area is output to the excel spreadsheet. If the user finds that the program is not satisfactorily measuring the contact areas, than the image histogram may be accessed through the COUNT window.

a. Select USER DEFINED tab in the COUNT window.
b. Select the THRESHOLD tab. When the window opens a mask appears over the current image. The vertical slider in the window may be moved horizontally to change the image threshold. As this is done, the magnitude of red in the image with either increase or decrease. Find the appropriate threshold level. Use the sequence control bar slider to view a few other images and confirm the new threshold level is appropriate. Close the window.
c. Rerun the macro. IMPORTANT: the default for the macro is to count dark objects. The user must select the user defined option while the macro is running to take advantage of the new threshold level.

14. After the areas are output to the Excel file, copy and paste them into the data output (time, load, and displacement data) obtained during the experiment. Since the displacement data is universally applied to each lens, a data file may be created for each lens in this manner.

Matlab

The Matlab programs require slightly more time to get running, but they are much more robust at handling images of varying intensity after the thresholds are optimized. These programs are recommended for difficult to analyze data or a multitude of lens data. There are two main programs: singlelens_bkchk and singlelens_area. The first program determines the centroid x,y position of the lens and the second program analyzes sequential images for the contact area, centroid, and eccentricity.

1. First open up the Matlab program and open the singlelens_bkchk and singlelens_area programs.
2. Copy the image files in .tif format into the appropriate directory (*c:\aaron\images*). Find the first image of the sequence (*00.tif*) and the maximum contact image. Copy these images to the appropriate directory (*c:\aaron\images\background*). The filenames must be changed in order for the Matlab program to recognize them. Change the first image filename to *background.tif* and the maximum contact area image to *backgroundmax.tif*.
3. Go to the command line window in Matlab. Type singlelens_bkchk to run the first program. The first image and maximum contact area image should open. Several other images will open in various forms as the program executes. The essential event to verify the program worked properly is that the maximum contact area is labeled at the end of program.

a. Singlelens_bkchk processes the two images to determine the centroid coordinates (x,y) of the maximum contact. These coordinates are used in the area analysis program discussed in the next step. The final image should show the contact areas with a unique number (colored in the figure) label.
b. If the program does not properly find the centroid of the contact area, please see the program notes in the Appendix to determine which values must be changed improve the analysis.

4. Type singlelens_area in the command window to run the second program. Several different figures will open and close as the program operates. Perhaps the most important figure is the histogram. As the lens comes into contact, the histogram will change from the background intensity to the background plus an additional signal from the dark contact area, see Figure 6. The location of the additional signal determines the threshold cut off for creating the binary image (see program notes in the Appendix).
5. A composite image is created during program execution and does not close after it is created. The image is titled *composite X* where the X represents the loop number. This image is an overlay of all of the contact areas at that point in time during the test. The colorbar shifts from blue to red, where blue is a short contact time and red is longer contact times (earlier contact). These images provide a colorful representation of the contact behavior of the lens.
6. After the program stops, the area data is output to the *c:\aaron\images* directory. The filenames are in the following format: lens#axbyc.txt, where a is the lens number determined from the singlelens_bkchk program, b is the centroid x coordinate for the contact area, and c is the centroid y coordinate for the contact area. The centroid values are important because they allow the user to cross reference the data file with the lens contact area in the composite image.
7. Open the .txt file with Excel to view the areas. The Matlab areas should be converted from pixels^2 to μm^2 using the conversion factors provided in Table 2.

   Table 2: Conversion factors for converting Matlab measured areas to metric units.

Magnification	Conversion (μm^2/pixel^2)	Std. Dev.
2.5	5.977	1.118
5	1.556	0.1464
10	0.4247	0.0172

8. After converting the area data copy and paste the areas into the data file to create a unique file for the lens.

Multiple contact area image analysis

1. First open up the Matlab program and open the multilens_bkchk and multilens_area programs.
2. Copy the image files in .tif format into the appropriate directory (*c:\aaron\images*). Find the first image of the sequence (*00.tif*) and the maximum contact image. Copy these images to the appropriate directory (*c:\aaron\images\background*). The filenames must be changed in order for the Matlab program to recognize them. Change the first image filename to *background.tif* and the maximum contact area image to *backgroundmax.tif*.
3. Go to the command line window in Matlab. Type multilens_bkchk to run the first program. The first image and maximum contact area image should open. Several other images, in various forms, will open and close as the program executes. The essential event to verify the program worked properly is that the maximum contact area is labeled with a unique number (label) at the end of program.
    a. multilens_bkchk processes the two images to determine the centroid coordinates (x,y) of the maximum contact for every lens. These coordinates are used in the area analysis program discussed in the next step. The final image should show all of the contact areas with a unique number (colored in the figure) label. Do not worry if all of the lenses are not found. With so many parallel adhesion experiments to analyze, it is preferable to have as many lenses as possible. The user should minimize large blocks of area that is not representative of a lens position.
    b. If the program does not properly find the centroid of the contact areas, please see the program notes in the Appendix to determine which values must be changed improve the analysis.
4. Type multilens_area in the command window to run the second program. Several different figures will open and close as the program operates. Perhaps the most important figure is the histogram. As the lens comes into contact, the histogram will change from the background intensity to the background plus an additional signal from the dark contact area, see Figure 6. The location of the additional signal determines the threshold cut off for creating the binary image (see program notes in the Appendix).
5. A composite image is created during program execution and does not close after it is created. The image is titled *composite X* where the X represents the loop number. This image is an overlay of all of the contact areas at that point in time during the test. The colorbar shifts from blue to red, where blue is a short contact time and red is longer contact times (earlier contact). These images provide a colorful, qualitative representation of the contact behavior of the lens.
6. After the program stops, the area data is output to the *c:\aaron\images* directory. The filenames are in the following format: lens#axbyc.txt,

where a is the lens number determined from the multilens_bkchk program, b is the centroid x coordinate for the contact area, and c is the centroid y coordinate for the contact area. The centroid values are important because they allow the user to cross reference the data file with the lens contact area in the composite image. For example, specific lenses may be analyzed by finding their centroid on the composite image and matching it to the area data output from the program.

7. Open the .txt file with Excel to view the contact areas for each lens. The Matlab areas should be converted from pixels^2 to μm^2 using the conversion factors provided in Table 3.

   Table 3: Conversion factors for converting Matlab measured areas to metric units.

Magnification	Conversion (pixel^2/μm^2)	Std. Dev.
2.5	5.977	1.118
5	1.556	0.1464
10	0.4247	0.0172

8. After converting the area data copy and paste the areas into the data file to create a unique file for each lens.

Data Analysis

The JKR theory is a modification of the Hertz equations of contact that takes into account the adhesive forces within the contact zone of two materials. The governing equations for this theory have been derived by several authors throughout the literature. We provide a brief overview to accommodate experimental analysis. The equations may be derived by considering an elastic hemisphere brought into contact with a planar substrate under load (see figure 7). In this geometry, the elastic hemisphere is defined by its modulus (E), Poisson's ratio (ν), and radius-of-curvature (R). The equilibrium contact area, A, between the hemisphere and the planar substrate may be described by considering an energy balance over the system.

The system energy is the sum of the elastic energy stored within the deformed hemisphere, U_E, the potential energy of the load, U_P, and the interfacial energy, U_S. The derivative of total free energy must be a minimum at contact equilibrium and we may take the derivative of the total free energy in the system to elicit two important terms: the change in potential energy of the system and the interfacial energy. The change in potential energy of the system is given by the potential energy of the load and the elastic energy stored within the hemisphere as a result of deformation as shown in equation (1)

$$\partial U_E / \partial A \Big|_P + \partial U_P / \partial A \Big|_P \qquad (1)$$

where the first term is the elastic energy within the hemisphere and the second term is the potential energy of the load. The interfacial energy is defined by

$$dU_S = -(\gamma_L + \gamma_S - \gamma_{LS})dA = -WdA \quad (2)$$

where γ_L is the surface energy of the lens, γ_S is the surface energy of the substrate, γ_{LS} is the interfacial energy between the lens and substrate, and W is the thermodynamic work of adhesion. The energy required to increase surface area during the loading curve of an axisymmetric adhesion test is bounded by the thermodynamic work of adhesion (W), also known as Dupre's energy of adhesion, which is the energy required to reversibly separate two joined surfaces into two free surfaces. For adhesion testing we are often interested with the additional energy required to drive the separation between the two surfaces. Therefore, we now focus on the energy release rate. The energy release rate ($\mathcal{G}$) is defined using equation (1) and represents the amount of energy required to change contact area a unit amount. The solution for the energy release rate from equation (1), in the limit of small contact area, is given by

$$\mathcal{G} = \frac{(P' - P)^2}{8\pi E^* a^3} \quad (3)$$

where $P' = 4E^*a^3$ is the Hertz prediction of load, E^* is the system modulus, and a is the contact radius. Equation (3) shows that the energy available for changing the contact radius is given by the difference between Hertz predictions (no adhesion) and the adhesive case. Equation (3) is valuable because it can be used to quantify velocity dependant adhesion processes. The system modulus, E^*, combines the contribution of both materials to the compliance of the system.

$$\frac{1}{E^*} = \frac{(1-\nu_L^2)}{E_L} + \frac{(1-\nu_S^2)}{E_S} \quad (4)$$

In equation (4), E_L and ν_L are the tensile modulus and Poisson's ratio for the lens; E_S and ν_S are the tensile modulus and Poisson's ratio for the substrate, respectively. For a rigid planar substrate and an elastic hemispherical lens the compliance results from the deformation of the lens. Therefore, the system modulus is dominated by the lens modulus and the substrate term in equation (4) is assumed to be zero. If the moduli of the lens material are known prior to the experiment, the system modulus may be calculated directly from equation (4).

If the moduli of the lens material are not known, then equation (5) is often used to determine the system modulus by fitting experimental load, displacement, and contact area data.

$$\delta_{JKR} = \frac{a^2}{R} + \frac{P}{2E^* a} \quad (5)$$

Equation (5) is important because it provides a second method, independent of adhesion energy, to experimentally determine the system modulus.

One potential challenge of the current multilens technique is the inability to measure load on each lens of the array in conjunction with individual lens contact

areas. In this case, equation (3) and equation (5) can not be used to calculate $\mathcal{G}$ without the load measurement. This challenge may be overcome by the use of the overall lens array displacement rather than the load and requires one to rearrange the energy release rate in terms of displacement rather than load, as was shown by several authors. The system compliance, *C*, may be substituted into equation (3) to give the energy release rate, $\mathcal{G}$, as a function of displacement, δ.

$$C = \frac{2}{E^* a} = \frac{d\delta}{dP} = \frac{\delta' - \delta}{P' - P} \tag{6}$$

$$\mathcal{G} = \frac{E^* \left(\delta' - \delta\right)^2}{2\pi a} \tag{7}$$

where δ' is the Hertzian displacement given by a^2/R. Equation (7) may be rearranged to solve for δ to give the displacement-dependent form of the JKR equation.

$$\delta = \delta_O + \delta' - \sqrt{\frac{2\pi a \mathcal{G}}{E^*}} \tag{8}$$

In equation (8), δ_0 is the displacement at initial contact between the lens and substrate. From equation (8), we see that the displacement predicted for the adhesive case is always less than predicted by the Hertzian displacement. The initial contact between the lens and substrate is not directly measured in an experiment. Images of the contact between the hemisphere and substrate are taken during an experiment, but the initial contact often occurs slightly before the first image of contact. Uncertainty in the displacement at initial contact will also lead to uncertainty in the measured $\mathcal{G}$. Therefore, δ_0 and $\mathcal{G}$ are used as parameters to fit equation (8) to experimentally measured contact areas and the overall lens array displacement.

The displacement-based measurements should permit the determination of the work of adhesion at each lens of the multi-lens array for elastic contact using equation (8). However, Deruelle et al. have shown that displacement-based adhesion experiments deviate from JKR theory more than corresponding load measurements due to the confinement of the lens. Since the lenses within the multilens array are much smaller than those used for single lens JKR tests, we expect finite size effects will affect our ability to model the experimental data using the JKR equation. Our analysis will concentrate on whether the multilens system is affected by finite size effects and whether the $\mathcal{G}$ measured with multilens experiments is comparable to single lens experiments.

APPENDIX

Appendix A: Equipment List

Labview v6.0
GPIB control/acquisition board
Leica DMIRE2 inverted microscope
Burleigh Instruments 3200 multi-axis controller
Burleigh Instruments IW-812 actuator
Philtec fiber optic-reflectance compensated displacement sensors
Sensotec 50g load cell
Entran load cell meter
Multi-axis tip/tilt stage
Mad City Labs multi-axis nanoposition stage

Appendix B: Image Pro Macros

```
Option Explicit
'*************************************************************************
'  The macro creates a plot of average intensities
'  of each frame in a sequence within an AOI
'  and sends this data to Excel
'  For Image-Pro Plus 4.0
'*************************************************************************
Sub JKR_2_5X()

        ret = IpSCalSelect("2.50 X")                    'Select Calibration of Objective (you will need to check
for proper units (um)

        ret = IpBlbShow(1)                              'Opens Count/Size Dialogue Box
        ret = IpBlbDelete()                             'Deletes Previous Counts
        ret = IpBlbSetAttr(BLOB_AUTORANGE, 1)           'Selects Autorange
        ret = IpBlbSetAttr(BLOB_BRIGHTOBJ, 0)  'Selects Autorange to Measure Dark Objects
        ret = IpBlbEnableMeas(BLBM_AREA, 1)             'Selects Measurement of Area

        ret = IpDcUpdate(DC_RESET)                      'Clears Previous Data in DC
        ret = IpDcShow(3)                               'Goes to Layout Tab in DC
        ret = IpDcSelect("Count_Size", "BLBM_AREA", 2)  'Selects Count/Size, Area, Max. in DC
        ret = IpDcShow(4)                               'Goes to Export Tab in DC
        ret = IpDde(DDE_SET, "append", "1")             'In DDE Options, selects Append data to bottom
  ret = IpDde(DDE_SET, "row", "1")              'In DDE Options, positions data to start in Row 1, Column 1
        ret = IpDde(DDE_SET, "col", "1")                'In DDE Options, positions data to start in Row 1, Column 1
        ret = IpDde(DDE_SET, "target", "C:\Program Files\Microsoft Office\Office\excel.exe")   'Pushes OK to set/save DDE
Options
        ret = IpDcShow(5)                               'Selects Options Tab in DC
        ret = IpDcSet(DC_LEFTCOL, 1)
        ret = IpDcSet(DC_AUTO, 1)                       'Selects Automatic Collection in DC
        ret = IpBlbShow(1)                              'Opens Count/Size Window

        Dim numfr As Long
 ret=IpSeqGet(SEQ_NUMFRAMES,numfr)              'Get number of frames in sequence
 If numfr = 1 Then
  ret = IpMacroStop("Not a sequence image", 0)
  Exit Sub
 End If
 ret = IpSeqPlay(SEQ_FFRA)                      'Go to the first frame
 Dim i As Integer                               'Definition of as loop variable
        For i=0 To numfr-1
        ret = IpBlbCount()                              'Initiates Count
        ret = IpBlbUpdate(0)
        ret = IpDcSet(DC_AUTO, 1)                       'Selects Automatic Collection in DC
        ret = IpSeqPlay(SEQ_NEXT)                       'Goes to Next Frame in Sequence
 Next i

 ret = IpDcSaveData("",  S_Y_AXIS + S_X_AXIS + S_DDE)   'Exports Data to Excel ACTIVE WINDOW

End Sub
```

```
Option Explicit
'************************************************************************
'  The macro creates a plot of average intensities
'  of each frame in a sequence within an AOI
'  and sends this data to Excel
'  For Image-Pro Plus 4.0
'************************************************************************
Sub JKR_5X()

        ret = IpSCalSelect("5 X")                                   'Select Calibration of Objective (you will need to check
for proper units (um)

        ret = IpBlbShow(1)                                              'Opens Count/Size Dialogue Box
        ret = IpBlbDelete()                                             'Deletes Previous Counts
        ret = IpBlbSetAttr(BLOB_AUTORANGE, 1)           'Selects Autorange
        ret = IpBlbSetAttr(BLOB_BRIGHTOBJ, 0) 'Selects Autorange to Measure Dark Objects
        ret = IpBlbEnableMeas(BLBM_AREA, 1)             'Selects Measurement of Area

        ret = IpDcUpdate(DC_RESET)                              'Clears Previous Data in DC
        ret = IpDcShow(3)                                               'Goes to Layout Tab in DC
        ret = IpDcSelect("Count_Size", "BLBM_AREA", 2)  'Selects Count/Size, Area, Max. in DC
        ret = IpDcShow(4)                                               'Goes to Export Tab in DC
        ret = IpDde(DDE_SET, "append", "1")             'In DDE Options, selects Append data to bottom
  ret = IpDde(DDE_SET, "row", "1")              'In DDE Options, positions data to start in Row 1, Column 1
        ret = IpDde(DDE_SET, "col", "1")                'In DDE Options, positions data to start in Row 1, Column 1
        ret = IpDde(DDE_SET, "target", "C:\Program Files\Microsoft Office\Office\excel.exe")   'Pushes OK to set/save DDE
Options
        ret = IpDcShow(5)                                               'Selects Options Tab in DC
        ret = IpDcSet(DC_LEFTCOL, 1)
        ret = IpDcSet(DC_AUTO, 1)                               'Selects Automatic Collection in DC
        ret = IpBlbShow(1)                                              'Opens Count/Size Window

        Dim numfr As Long
 ret=IpSeqGet(SEQ_NUMFRAMES,numfr)              'Get number of frames in sequence
 If numfr = 1 Then
  ret = IpMacroStop("Not a sequence image", 0)
  Exit Sub
 End If
 ret = IpSeqPlay(SEQ_FFRA)                              'Go to the first frame
 Dim i As Integer                                               'Definition of as loop variable
        For i=0 To numfr-1
        ret = IpBlbCount()                                              'Initiates Count
        ret = IpBlbUpdate(0)
        ret = IpDcSet(DC_AUTO, 1)                               'Selects Automatic Collection in DC
        ret = IpSeqPlay(SEQ_NEXT)                               'Goes to Next Frame in Sequence
 Next i

 ret = IpDcSaveData("",  S_Y_AXIS + S_X_AXIS + S_DDE)    'Exports Data to Excel ACTIVE WINDOW

End Sub
```

```
Option Explicit
'************************************************************************
'  The macro creates a plot of average intensities
'  of each frame in a sequence within an AOI
'  and sends this data to Excel
'  For Image-Pro Plus 4.0
'************************************************************************
Sub JKR_10X()

        ret = IpSCalSelect("10 X")                          'Select Calibration of Objective (you will need to check
for proper units (um)

        ret = IpBlbShow(1)                                  'Opens Count/Size Dialogue Box
        ret = IpBlbDelete()                                 'Deletes Previous Counts
        ret = IpBlbSetAttr(BLOB_AUTORANGE, 1)           'Selects Autorange
        ret = IpBlbSetAttr(BLOB_BRIGHTOBJ, 0) 'Selects Autorange to Measure Dark Objects
        ret = IpBlbEnableMeas(BLBM_AREA, 1)             'Selects Measurement of Area

        ret = IpDcUpdate(DC_RESET)                          'Clears Previous Data in DC
        ret = IpDcShow(3)                                   'Goes to Layout Tab in DC
        ret = IpDcSelect("Count_Size", "BLBM_AREA", 2)  'Selects Count/Size, Area, Max. in DC
        ret = IpDcShow(4)                                   'Goes to Export Tab in DC
        ret = IpDde(DDE_SET, "append", "1")             'In DDE Options, selects Append data to bottom
  ret = IpDde(DDE_SET, "row", "1")              'In DDE Options, positions data to start in Row 1, Column 1
        ret = IpDde(DDE_SET, "col", "1")                'In DDE Options, positions data to start in Row 1, Column 1
        ret = IpDde(DDE_SET, "target", "C:\Program Files\Microsoft Office\Office\excel.exe")   'Pushes OK to set/save DDE
Options
        ret = IpDcShow(5)                                   'Selects Options Tab in DC
        ret = IpDcSet(DC_LEFTCOL, 1)
        ret = IpDcSet(DC_AUTO, 1)                           'Selects Automatic Collection in DC
        ret = IpBlbShow(1)                                  'Opens Count/Size Window

        Dim numfr As Long
 ret=IpSeqGet(SEQ_NUMFRAMES,numfr)                  'Get number of frames in sequence
 If numfr = 1 Then
  ret = IpMacroStop("Not a sequence image", 0)
  Exit Sub
 End If
 ret = IpSeqPlay(SEQ_FFRA)                          'Go to the first frame
 Dim i As Integer                                       'Definition of as loop variable
        For i=0 To numfr-1
        ret = IpBlbCount()                                  'Initiates Count
        ret = IpBlbUpdate(0)
        ret = IpDcSet(DC_AUTO, 1)                           'Selects Automatic Collection in DC
        ret = IpSeqPlay(SEQ_NEXT)                           'Goes to Next Frame in Sequence
 Next i

 ret = IpDcSaveData("",  S_Y_AXIS + S_X_AXIS + S_DDE)   'Exports Data to Excel ACTIVE WINDOW

End Sub
```

Appendix C: Matlab Software Code

Single lens Analysis

Single Lens BKCHK

```
% function [outargs]=nist1(o,inargs)
%   [outargs]=nist1(o,inargs) here describe in more detail
% the purpose of this routine is to compute the contact area of the
% lenses in the tif-images

%open background image
I = imread('c:\aaron\images\background\background.tif');
%open maximum contact image.
J = imread('c:\aaron\images\background\backgroundmax.tif');
J=double(J);
I=double(I);

%show both images.
figure;imshow(I);title('Background');
figure;imshow(J);title('Background Max');
pause
%blur with a gaussian filter
PSF = fspecial('gaussian',32,10);
Blurred = imfilter(J,PSF,'symmetric','conv');
%subtract blurred image from original to normalize intensity
subt=J-Blurred;
figure; imshow(subt);title('subtracted image after blur');
%create an binary image of the subtracted image
pause
subt = double(subt);
%figure; imshow(subt);title('double subtracted image');
%show the historgram
figure;hist(subt);
pause;
%create the binary image
%create the binary image
im3=ones(size(subt));
   idy=find(0<subt<-10);
   idx=find(subt>0);
   idz=find(subt<-10);
   im3(idy)=1;
   im3(idx)=0;
   im3(idz)=0;
%convert to binary image
   im3=logical(im3);
```

```
figure; imshow(im3);title('binary image');
pause
%filter image to remove small particles
se = strel('disk',2);
fillholes = imopen(im3, se);
figure; imshow(fillholes); title('imopen');
%dilate to hopefully connect lens boundaries
SE1 = ones(5,5);
fillholes = imdilate(fillholes,SE1);
figure; imshow(im3); title('dilate');
pause
%erode edge of lenses to prevent them from contacting each other
se3 = strel('disk',1);
fillholes = imerode(fillholes, se3);
figure; imshow(fillholes); title('erode');
pause
%remove larger dust particles
fillholes = bwareaopen(fillholes, 15000,8);
fillholes=bwfill(fillholes,'holes',8);
figure; imshow(fillholes); title('area open');
%clean up the area around the lens areas
se2 = strel('disk',20);
fillholes = imopen(fillholes, se2);
figure; imshow(fillholes); title('disk 40');
pause
%erode edge of lenses to prevent them from contacting each other
se3 = strel('disk',3);
fillholes = imerode(fillholes, se3);
figure; imshow(fillholes); title('erode');
pause
%label the objects within the image
LW=bwlabel(fillholes,8);
%collect the statistics of the objects
stats1 = regionprops(LW, 'Centroid', 'Area');
figure; imagesc(LW); title('smoothed-labeled image'); colorbar; colormap('jet');
pause
%collect the statistics for the centroid of the lens in view.
bkcent = [stats1.Centroid];
bkarea = [stats1.Area];
bkcentel = numel(bkcent)/2;
bkcentxy = reshape(bkcent,2,bkcentel);
bkcentxyr = bkcentxy.';
pause
close all
end
```

Single Lens AREA

```
% the purpose of this routine is to compute the contact area of the
% mircolenses in the tif-images

%open the background image
pth2='C:\aaron\images\background';
bgname=fullfile(pth2,'background.tif');
im_bg=double(imread(bgname,'tif'));
%figure; imagesc(im_bg); title('background image'); colorbar; colormap('gray');

% imagesc(im_bg);
pth='C:\aaron\images\'
o.recursive=0; o.ask=0; filter='*.tif';
flist1=dir(fullfile(pth,'*.tif'));
%read the names of all the images into flist{i}
for i=1:length(flist1);
   flist{i}=fullfile(pth,flist1(i).name);
end
%set up data arrays to hold results
numimage=length(flist);
temp=bkcentel*5;
areastatistics=zeros([numimage temp]);
%create a blank image.
comp1=zeros(size(im_bg));
%loop through all of the images in the directory
for ij=1:length(flist)-1;
   fname=flist{ij};
   %best way I have found to sort the files
   loop=int2str(ij-1);
   %work through files in numerical name order.
   fname2= [pth, loop];
   fname3= [fname2, '0.tif']
 %read in the original image.
   im=double(imread(fname3,'tif'));
%show the original image
  figure; imagesc(im); title('uncorrected image'); colorbar; colormap('gray');
 pause(5);
   close;
%subtract the background
   im2=im-im_bg;
   figure; imagesc(im2); title('bg corrected image'); colorbar;colormap('gray');
   pause(3);
   close;
%show histogram
   figure; hist(im2(:),25);title('im-im_bg histo');
   pause(3);
```

```
    close;
%create the binary image
    im3=zeros(size(im));
%set the threshold
    idx=find(im2<-38);
    im3(idx)=1;
    figure; imagesc(im3); title('thresholded image'); colorbar;colormap('gray');
    pause(3);
    close;
%fill holes in the image
im4=bwfill(im3,'holes',8);
figure; imagesc(im4); title('holes filled'); colorbar; colormap('gray');
pause(3);
 close;
%clean up the area around the lens areas
se2 = strel('disk',30);
im5 = imopen(im4, se2);
%figure; imshow(im5); title('disk 2');
%label the lens
im6 = bwlabel(im5, 8);
%collect the lens properties
stats = regionprops(im6, 'Area', 'Centroid', 'EquivDiameter', 'Eccentricity');
%close;
figure; imagesc(im6); title('remove dust'); colorbar; colormap('jet');
 pause(3);
 close;
%create the composite image
%pause
%close
composite=im3;
comp1=imadd(composite, comp1);
R=rem((ij-1), 20);
if R==0;
gtitle=['composite  ', loop];
%show the composite image
figure; imagesc(comp1); title(gtitle); colorbar; colormap('jet');
pause(5);
end
%collect the centroid of the lens from bkchk program
for i=1:bkcentel;
c = int16(bkcentxyr(i, 1));
r =int16(bkcentxyr(i, 2));

aa=im6(r,c);
%write the statistics of the lens to a file.
if aa==0;
```

```
    areastatistics(ij, (i-1)*5+1)=0;
    areastatistics(ij, (i-1)*5+ 2)=-1;
    areastatistics(ij, (i-1)*5+ 3)=-1;
    areastatistics(ij, (i-1)*5+ 4)=0;
    areastatistics(ij, (i-1)*5+ 5)=0;

else
    areastatistics(ij, (i-1)*5+ 1)=stats(aa).Area;
    areastatistics(ij, (i-1)*5+ 2)=stats(aa).Centroid(1);
    areastatistics(ij, (i-1)*5+ 3)=stats(aa).Centroid(2);
    areastatistics(ij, (i-1)*5+ 4)=stats(aa).EquivDiameter;
    areastatistics(ij, (i-1)*5+ 5)=stats(aa).Eccentricity;

end
end

end
%sort through lenses
lensstats=zeros([numimage 5]);
headers= strvcat('Area', 'Centroid(1)', 'Centroid(2)', 'EquivDiameter','Eccentricity');
for i=1:bkcentel
%sort through time or number of images
for ij=1:length(flist);
    lensstats(ij, 1)=areastatistics(ij, (i-1)*5+1);
    lensstats(ij, 2)=areastatistics(ij, (i-1)*5+ 2);
    lensstats(ij, 3)=areastatistics(ij, (i-1)*5+ 3);
    lensstats(ij, 4)=areastatistics(ij, (i-1)*5+ 4);
    lensstats(ij, 5)=areastatistics(ij, (i-1)*5+ 5);
end
%write data for each lens to the file
    lensnumber = int2str(i);
    x = num2str(max(lensstats(:,2)));
    y = num2str(max(lensstats(:,3)));
    filewrite = ['c:\aaron\images\lens#',lensnumber,'x', x, 'y', y,'.txt'];
    fid = fopen(filewrite,'a');
    fprintf(fid,'Area\tCentroid(1)\tCentroid(2)\tEquivDiameter\tEccentricity\n');
    fprintf(fid,'%E\t%E\t%E\t%E\t%E\n',lensstats');
    fclose(fid);
end
    lensstats=zeros([numimage 5]);
end
```

Multiple lens Analysis

Muliple Lens BKCHK

```
% the purpose of this routine is to compute the contact area of the
% mircolenses in the tif-images

%open background image
I = imread('c:\aaron\images\background\background.tif');
figure;imshow(I);title('Original Image');
%blur with a gaussian filter
PSF = fspecial('gaussian',32,10);
Blurred = imfilter(I,PSF,'symmetric','conv');
%show blurred image
figure;imshow(Blurred);title('Blurred Image');
I=double(I);
Blurred=double(Blurred);
%subtract blurred image from original to normalize intensity
subt=I-Blurred;
%show subtracted image
figure; imshow(subt);title('subtracted image');
%create a binary image of the subtracted image
im3=ones(size(subt));
   idx=find(subt>0);
   idy=find(subt<0);
   im3(idx)=0;
   im3(idy)=1;
%convert to binary image
   im3=logical(im3);
%show binary image
figure; imshow(im3);title('binary image');
%filter image to remove small particles; adjust size of this disk to clean more
%or less from the image.
se = strel('disk',1) ;
fillholes = imopen(im3, se);
figure; imshow(fillholes); title('imopen');
%dilate each pixel to connect lens boundaries;
SE1 = ones(2,2);
fillholes = imdilate(fillholes,SE1);
figure; imshow(fillholes); title('dilate');
pause
%erode edge of lenses to prevent them from contacting each other
%reduce some of the effects of dilation
se3 = strel('disk',4);%3
fillholes = imerode(fillholes, se3);
figure; imshow(fillholes); title('erode');
pause
```

```
%remove larger dust particles from the image…should not remove the lenses
fillholes = bwareaopen(fillholes, 15000,8);
fillholes=bwfill(fillholes,'holes',8);
figure; imshow(fillholes); title('area open');
%clean up the area around the lens areas
se2 = strel('disk',5);
fillholes = imopen(fillholes, se2);
figure; imshow(fillholes); title('disk 40');
pause
%erode edge of lenses to prevent them from contacting each other
se3 = strel('disk',12);
fillholes = imerode(fillholes, se3);
figure; imshow(fillholes); title('erode');
pause
%label the lenses within the image
LW=bwlabel(fillholes,8);
stats1 = regionprops(LW, 'Centroid', 'Area');
figure; imagesc(LW); title('smoothed-labeled image'); colorbar; colormap('jet');
pause
%write the centroid of each lens to the data file.
bkcent = [stats1.Centroid];
bkarea = [stats1.Area];
bkcentel = numel(bkcent)/2;
bkcentxy = reshape(bkcent,2,bkcentel);
bkcentxyr = bkcentxy.';
pause
close all
end
```

Multiple Lens AREA

```
% the purpose of this routine is to compute the contact area of the
% mircolenses in the tif-images
% open background image
pth2='C:\aaron\images\background';
bgname=fullfile(pth2,'background.tif');
im_bg=double(imread(bgname,'tif'));
% show background image
%figure; imagesc(im_bg); title('background image'); colorbar; colormap('gray');

pth='C:\aaron\images\'
o.recursive=0; o.ask=0; filter='*.tif';
flist1=dir(fullfile(pth,'*.tif'));
%create a path name for each image
for i=1:length(flist1);
   flist{i}=fullfile(pth,flist1(i).name);
end
numimage=length(flist);
%set up data arrays
% bkcentel is the number of lenses found in multilens_bkchk
temp=bkcentel*5;
areastatistics=zeros([numimage temp]);
comp1=zeros(size(im_bg));
%begin loop to analyzed each image
for ij=1:length(flist);
   fname=flist{ij};
   %work through image files in numerical name order.
   fname2= [pth, loop];
%zero added to name for compatibility to Labview output
   fname3= [fname2, '0.tif']
%read in the image file
   im=double(imread(fname3,'tif'));
   %figure; imagesc(im); title('uncorrected image'); colorbar; colormap('gray');
%subtract background
   im2=im-im_bg;
   %figure; imagesc(im2); title('bg corrected image'); colorbar;colormap('gray');
%show histogram
   %figure; hist(im2(:),25);
   im3=zeros(size(im));
%set image threshold
   idx=find(im2<-10);
%create binary image
   im3(idx)=1;
%show binary image
```

```
   %figure; imagesc(im3); title('thresholded image'); colorbar;colormap('gray');
%create a solid contact area instead of a ring
%if contact areas are not recognized, may need to add a dilate step.
im4=bwfill(im3,'holes',8);
%show filled figure
%figure; imagesc(im4); title('holes filled'); colorbar; colormap('gray');
%use a structure element to clean up the noise area around each contact area
se2 = strel('disk',1);
im5 = imopen(im4, se2);
%show opened image
%figure; imshow(im5); title('disk 2');
%label all structures (contact areas) within the image
im6 = bwlabel(im5, 8);
%calculate the statistics of each structure
stats = regionprops(im6, 'Area', 'Centroid', 'EquivDiameter', 'Eccentricity');
%show all labeled structures
%figure; imagesc(im6); title('remove dust'); colorbar; colormap('jet');
%create the composite image
composite=im3;
comp1=imadd(composite, comp1);
R=rem((ij-1), 20);
if R==0;
gtitle=['composite  ', loop];
%show composite image
figure; imagesc(comp1); title(gtitle); colorbar; colormap('jet');xlabel('Pixels');
ylabel('Pixels');
pause(5);
end

%obtain the coordinates for the centroid of each lens: determined from multilens_bkchk
for i=1:bkcentel;
c = int16(bkcentxyr(i, 1));
r =int16(bkcentxyr(i, 2));
%write the statistics for each lens from this single image to a data array.
aa=im6(r,c);
if aa==0;
   areastatistics(ij, (i-1)*5+1)=0;
   areastatistics(ij, (i-1)*5+ 2)=-1;
   areastatistics(ij, (i-1)*5+ 3)=-1;
   areastatistics(ij, (i-1)*5+ 4)=0;
   areastatistics(ij, (i-1)*5+ 5)=0;

else
   areastatistics(ij, (i-1)*5+ 1)=stats(aa).Area;
   areastatistics(ij, (i-1)*5+ 2)=stats(aa).Centroid(1);
   areastatistics(ij, (i-1)*5+ 3)=stats(aa).Centroid(2);
```

```
   areastatistics(ij, (i-1)*5+ 4)=stats(aa).EquivDiameter;
   areastatistics(ij, (i-1)*5+ 5)=stats(aa).Eccentricity;

end
end

end
%collect data from all images and write it to a file
%sort through lenses
lensstats=zeros([numimage 5]);
headers= strvcat('Area', 'Centroid(1)', 'Centroid(2)', 'EquivDiameter','Eccentricity');
for i=1:bkcentel
%sort through time or number of images
for ij=1:length(flist);
   lensstats(ij, 1)=areastatistics(ij, (i-1)*5+1);
   lensstats(ij, 2)=areastatistics(ij, (i-1)*5+ 2);
   lensstats(ij, 3)=areastatistics(ij, (i-1)*5+ 3);
   lensstats(ij, 4)=areastatistics(ij, (i-1)*5+ 4);
   lensstats(ij, 5)=areastatistics(ij, (i-1)*5+ 5);
end

%write data for each lens to the file
   lensnumber = int2str(i);
   x = num2str(max(lensstats(:,2)));
   y = num2str(max(lensstats(:,3)));
   filewrite = ['c:\aaron\images\lens#',lensnumber,'x', x, 'y', y,'.txt'];
   fid = fopen(filewrite,'a');
   fprintf(fid,'Area\tCentroid(1)\tCentroid(2)\tEquivDiameter\tEccentricity\n');
   fprintf(fid,'%E\t%E\t%E\t%E\t%E\n',lensstats');
   fclose(fid);
end
   lensstats=zeros([numimage 5])
end
```